Computazione per l'interazione naturale: classificazione probabilistica

Corso di Interazione uomo-macchina II

Prof. Giuseppe Boccignone

Dipartimento di Informatica
Università di Milano

boccignone@di.unimi.it
http://boccignone.di.unimi.it/IUM2_2014.html

Classificazione probabilistica

- Predire il genere dall'altezza:

Dati osservati (likelihood)

$p(h|C=1) \qquad p(h|C=0)$

Prob. a priori

$P(C=1) \qquad P(C=0)$

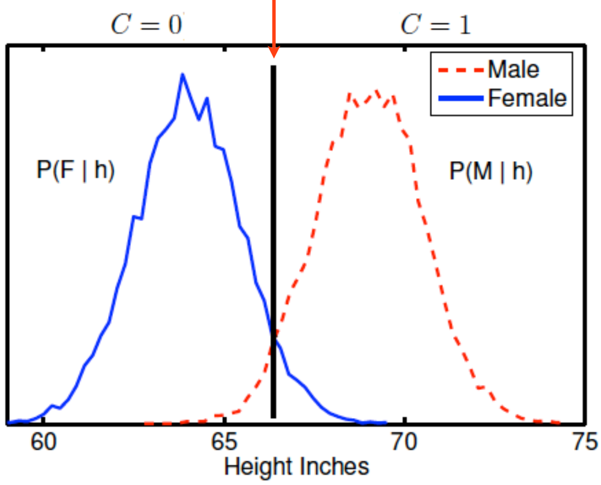
Prob. a posteriori (Bayes)

$$\boxed{P(C=1|h) = \frac{p(h|C=1)P(C=1)}{p(h)}}$$

$$p(h) = p(h|C=1)P(C=1) + p(h|C=0)P(C=0)$$

Regione di separazione

$P(C=1|h) = P(C=0|h)$



Classificazione probabilistica

- Date le probabilità a posteriori:

$$P(C=1|h) = \frac{p(h|C=1)P(C=1)}{p(h)}$$

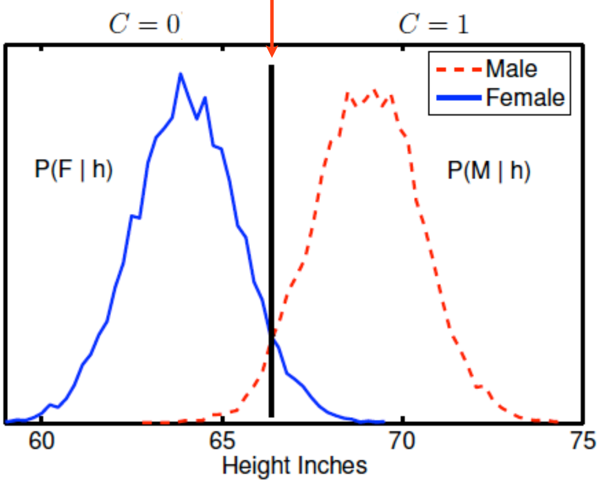
- Classifichiamo $C=1$ se:

$$P(C=1|h) > P(C=0|h)$$

- Possiamo
  - Trovare f : X → {1, . . . ,K} (**funzione discriminante**) che mappa ogni input x in una classe C_i (con i = f(x))

Regione di separazione

$P(C=1|h) = P(C=0|h)$



Classificazione probabilistica

- Classifichiamo $C=1$ se:

$$P(C=1|h) > P(C=0|h)$$

- Possiamo
  - Trovare f : X → {1, . . . ,K} (**funzione discriminante**) che mappa ogni input x in una classe C_i (con i = f(x))
  - Esempio:

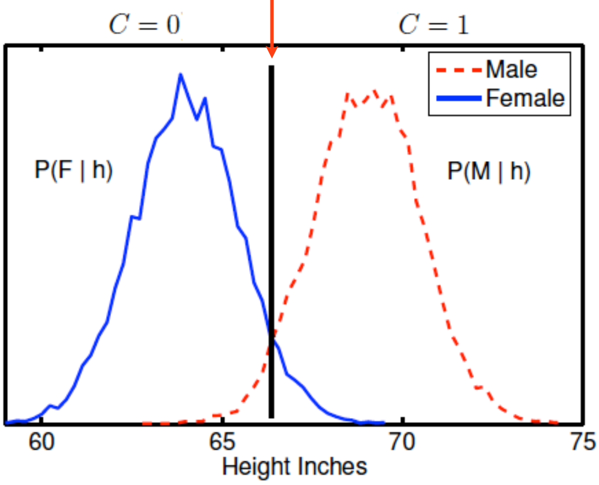
$$f(h) = \log \frac{P(C=1|h)}{P(C=0|h)}$$

$f(h) > 0 \rightarrow C=1$ (male)

$f(h) < 0 \rightarrow C=0$ (female)

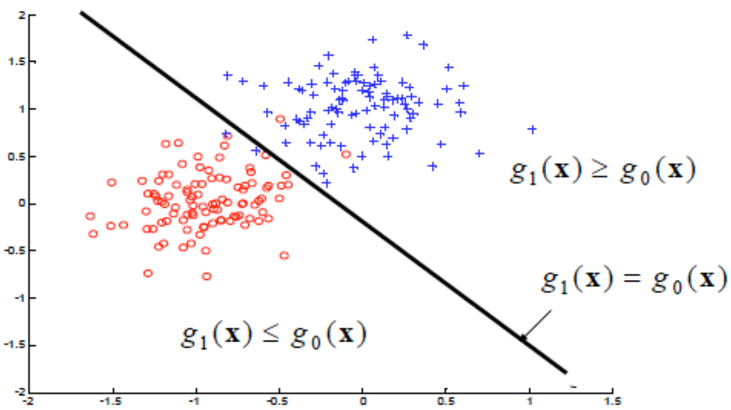
Regione di separazione

$P(C=1|h) = P(C=0|h)$



Metodologia generale: modelli discriminativi

- Discriminativi non probabilistici:
  - Trovare f : X → {1, . . . ,K} (**funzione discriminante**) che mappa ogni input x in una classe C_i (con i = f(x))
  - Esempio: SVM (Support Vector Machine)



Funzioni di discriminazione //lineari e lineari generalizzate

- Cos'è un classificatore lineare?
  - La classificazione è intrinsecamente non lineare
- Semplicemente: la parte adattiva del classificatore (ad esempio i pesi) è lineare (come per la regressione)

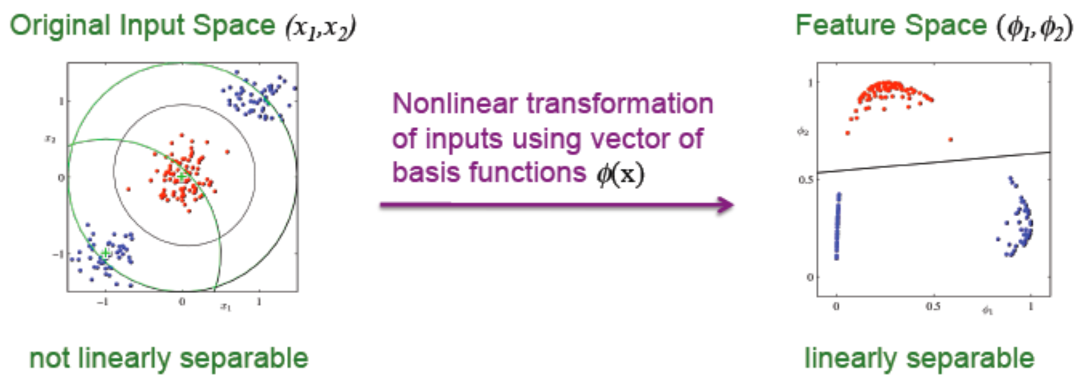
$$z(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 \qquad Decision = f(z(\mathbf{x}))$$

parte adattiva lineare — decisione non lineare

- Casi possibili:
  - non linearità fissata a valle della parte adattiva (decisione sigmoidale)
  - non linearità fissata a monte della parte adattiva (funzioni di base non lineari)

Funzioni di discriminazione
//lineari e lineari generalizzate

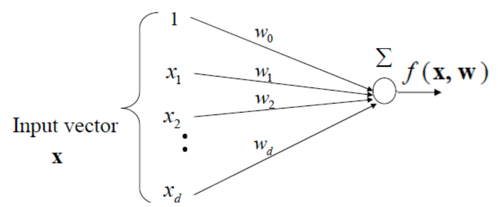
- non linearità fissata a monte della parte adattiva (funzioni di base non lineari)



Funzioni di discriminazione
//lineari e lineari generalizzate

- Consentono di assegnare ogni input x a una classe
- Definiscono una partizione dello spazio degli input in regioni R_i tali che se $x \in R_i$ allora x viene assegnato alla classe C_i
- Modello lineare:

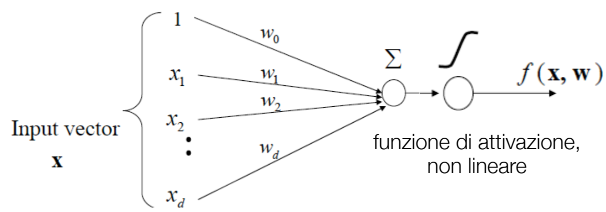
$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$



- Modello lineare generalizzato (GLM):

$$y(\mathbf{x}) = a(\mathbf{w}^T\mathbf{x} + w_0)$$

funzione di attivazione, non lineare

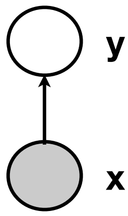


Metodologia generale: modelli discriminativi

- Discriminativi non probabilistici:
  - Trovare f : X → {1, . . . ,K} (**funzione discriminante**) che mappa ogni input x in una classe C_i (con i = f(x))

già visti

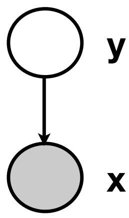
- Discriminativi probabilistici:
  - Effettuare direttamente una stima di p(**y** | **x**, T) dal training set
    - questo approccio è detto discriminativo, perchè, a partire da T , viene derivata una caratterizzazione dell'output in funzione delle features, in modo tale da discriminare, dato un elemento, il più probabile tra i possibili valori dell'output



  - Esempio: regressione logistica (LR)

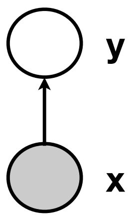
Metodologia generale: modelli generativi

- In un approccio generativo, viene derivato, per ogni possibile output, un modello (sotto forma di distribuzione di probabilità) degli elementi associati a quell'output
  - Descrizione completa della situazione: distribuzione di probabilità congiunta p(**x**, **y** | T), derivata a partire dal training set



$$p(\mathbf{x}, \mathbf{y} \mid T) = p(\mathbf{y} \mid \mathbf{x}, T)\, p(\mathbf{x} \mid T)$$

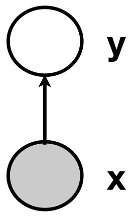
  - Inferire la probabilità a posteriori mediante regola di Bayes



$$p(\mathbf{y} \mid \mathbf{x}, T) = p(\mathbf{x}, \mathbf{y} \mid T) / p(\mathbf{x} \mid T)$$

Modelli discriminativi probabilistici

- Discriminativi probabilistici:
    - Effettuare direttamente una stima di p(**y** | **x**, T) dal training set
        - questo approccio è detto discriminativo, perchè, a partire da T , viene derivata una caratterizzazione dell'output in funzione delle features, in modo tale da discriminare, dato un elemento, il più probabile tra i possibili valori dell'output



Modelli discriminativi probabilistici //Regressione Logistica

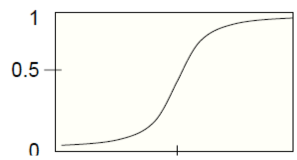
- Trasformiamo il dato di input $\mathbf{x} = [x_1, \cdots, x_D]^\mathsf{T}$ usando M funzioni di base

$$\phi(\mathbf{x}) = [\phi_1(\mathbf{x}, \cdots, \phi_M(\mathbf{x})]^\mathsf{T},$$

- Usiamo un modello lineare per descrivere la log-likelihood ratio

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \mathbf{w}^\mathsf{T}\phi(\mathbf{x})$$

Funzione Logistica

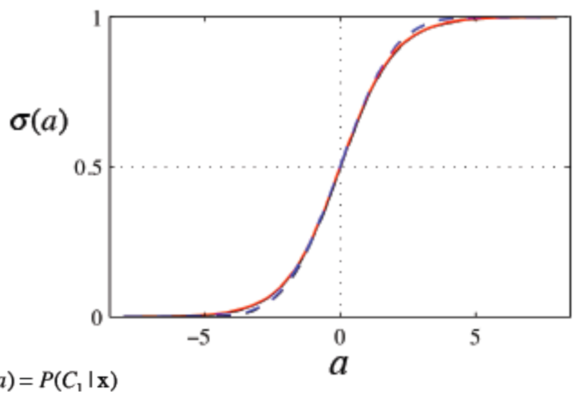


- Poichè $P(C=1|\mathrm{x}) + P(C=0|\mathrm{x}) = 1$

$$\frac{P(C=1|\mathbf{x})}{1-P(C=1|\mathbf{x})} = \exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right) \Rightarrow \boxed{\begin{aligned} P(C=1|\mathbf{x}) &= \frac{1}{1+\exp\left(-\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right)} \\ &= \frac{\exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right)}{1+\exp\left(\mathbf{w}^\mathsf{T}\phi(\mathbf{x})\right)} \end{aligned}}$$

Modelli discriminativi probabilistici //regressione logistica: funzione logistica

Logistica $$\sigma(a) = \frac{1}{1+\exp(-a)}$$



Proprietà di simmetria $$\sigma(-a) = 1 - \sigma(a)$$

La funzione inversa è la funzione logit $$a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

$$\sigma(a) = P(C_1 \mid \mathbf{x})$$
$$\downarrow$$
$$a = \ln\left(\frac{\sigma}{1-\sigma}\right) \Rightarrow \ln[p(C_1|\mathrm{x})/p(C_2|\mathrm{x})]$$ Log-odds ratio

Derivata $$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

Modelli discriminativi probabilistici //Regressione Logistica

- Algoritmo di base:
- Step 1. Calcolo la funzione logit $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$ con una regressione

$$\boxed{\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})}} = \mathbf{w}^{\mathsf{T}}\phi(\mathbf{x})$$

- Step 2. Inverto la logit ottenendo la logistica, cioè la posteriori

$$P(C=1|\mathbf{x}) = \frac{1}{1+\exp\left(-\mathbf{w}^{\mathsf{T}}\phi(\mathbf{x})\right)}$$

Modelli discriminativi probabilistici //regressione logistica: esempio a 2 classi

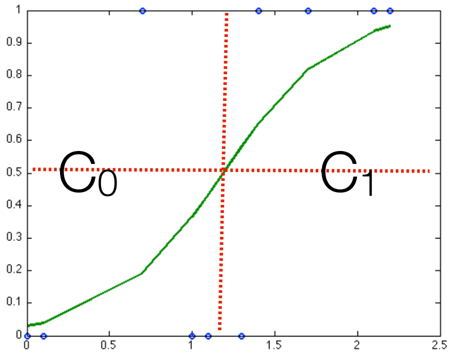
```
function EsempioLogisticRegression()
    %dati di training
    x = [0.0 0.1 0.7 1.0 1.1 1.3 1.4 1.7 2.1 2.2]';
    y = [0 0 1 0 0 0 1 1 1 1]';

    %fitting con generalized linear model dello Statistical
    %Toolbox

    w = glmfit(x,[y ones(10,1)],'binomial','link','logit')


    %predizione lineare
    %z = w(1) + x * (w(2))

    %applicazione della funzione logistica alla componente
    %lineare
    z = Logistic(w(1) + x * (w(2)))
    figure(1)
    plot(x,y,'o', x,z,'-', 'LineWidth',2)

end

function Output = Logistic(Input)

    Output = 1 ./ (1 + exp(-Input));

end
```

$$\log \frac{p(C_1 \mid \mathbf{x})}{p(C_0 \mid \mathbf{x})} = \mathbf{w}^T \mathbf{x} + w_0$$

$$p(C_1 \mid \mathrm{x}) = \frac{1}{1+\exp(-a)} = \sigma(a)$$

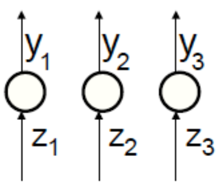
$$a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$



Modelli discriminativi probabilistici //regressione logistica

- Estensione a più classi: uso la decisione con funzione softmax



outputs $\quad y_i = \dfrac{e^{z_i}}{\sum_j e^{z_j}}$

$$\frac{\partial y_j}{\partial z_i} = \delta(i,j) y_j - y_j y_i$$

- La logistica è un caso particolare di softmax a due classi

$$y_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_0}} = \frac{1}{1+e^{-(z_1 - z_0)}}$$

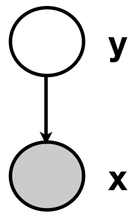
$z' = z_1 - z_0 = w_1 x - w_2 x = wx$

Modelli generativi di classificazione

- In un approccio generativo, viene derivato, per ogni possibile output, un modello (sotto forma di distribuzione di probabilità) degli elementi associati a quell'output

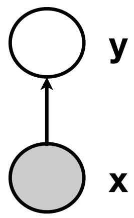
    - Descrizione completa della situazione: distribuzione di probabilità congiunta p(**x**, **y** | T), derivata a partire dal training set

$$p(\mathbf{x}, \mathbf{y} \mid T) = p(\mathbf{y} \mid \mathbf{x}, T)\, p(\mathbf{x} \mid T)$$



    - Inferire la probabilità a posteriori mediante regola di Bayes

$$p(\mathbf{y} \mid \mathbf{x}, T) = p(\mathbf{x}, \mathbf{y} \mid T) \,/\, p(\mathbf{x} \mid T)$$



Modelli generativi di classificazione

- Prima (non generativo)

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \mathbf{w}^{\mathsf{T}} \phi(\mathbf{x})$$

modello diretto della pdf a posteriori

- Adesso definiamo una funzione discriminante che tiene conto degli a priori sulle classi

modello della distribuzione a priori

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \log \frac{P(\mathbf{x}|C=1)P(C=1)}{P(\mathbf{x}|C=0)P(C=0)}$$

modello della likelihood

Modelli generativi di classificazione

- Adesso definiamo una funzione discriminante che tiene conto degli a priori sulle classi

modello della distribuzione a priori

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \log \frac{P(\mathbf{x}|C=1)P(C=1)}{P(\mathbf{x}|C=0)P(C=0)}$$

- Un'opzione semplice: dato il training set, conto il numero di target appartenenti alla classe k

$$\widehat{P}(C=k) = \frac{1}{N_k}\sum_{n=1}^{N}\delta(t_n, k)$$

Modelli generativi di classificazione // Modello Gaussiano (GDA)

- Adesso definiamo una funzione discriminante che tiene conto degli a priori sulle classi

$$\log \frac{P(C=1|\mathbf{x})}{P(C=0|\mathbf{x})} = \log \frac{P(\mathbf{x}|C=1)P(C=1)}{P(\mathbf{x}|C=0)P(C=0)}$$

modello della likelihood

- Funzione di verosimiglianza Gaussiana:

$$p(\mathbf{x}|C=k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathsf{T}}\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

Modelli generativi di classificazione
// Modello Gaussiano (GDA)

$$p(\mathbf{x}|C=k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\mathsf{T}\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

```
%% Stima dei parametri (ML)
class_var = [];
for c = 1:length(cl)
    pos = find(t==cl(c));
    % Medie
    class_mean(c,:) = mean(X(pos,:));
    % Matrici di covarianza
    class_var(:,:,c) = cov(X(pos,:),1);
end


%% Probabilità predittive
[Xv,Yv] = meshgrid(-3:0.1:7,-6:0.1:6);
Probs = [];
for c = 1:length(cl)
    temp = [Xv(:)-class_mean(c,1) Yv(:)-class_mean(c,2)];
    sigma_k = class_var(:,:,c);
    const = -log(2*pi) - log(det(sigma_k));
    Probs(:,:,c) = reshape(exp(const - 0.5*diag(temp*inv(sigma_k)*temp')),size(Xv));;
end

Probs = Probs./repmat(sum(Probs,3),[1,1,3]);
```

Modelli generativi di classificazione
// Modello Gaussiano (GDA)

- Che forma ha la funzione discriminante (la curva di separazione?

$$\log \frac{P(C=k|\mathbf{x})}{P(C=l|\mathbf{x})} = \log \frac{P(\mathbf{x}|C=k)}{P(\mathbf{x}|C=l)} + \log \frac{P(C=k)}{P(C=l)}$$

$$= \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + \mathbf{w}^\mathsf{T}\mathbf{x} + b_0$$

Forma quadratica

$$\mathbf{A} = \Sigma_l^{-1} - \Sigma_k^{-1}$$

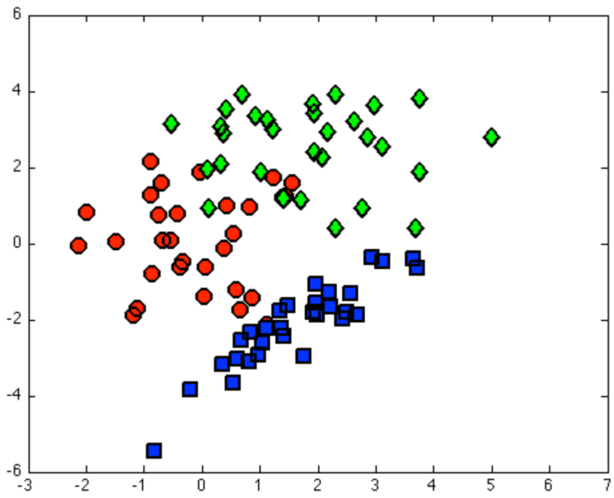
Forma lineare: se le matrici di covarianza sono uguali $\mathbf{w}^\mathsf{T}\mathbf{x} + b_0$

$$\mathbf{w} = \Sigma_k^{-1}\boldsymbol{\mu}_k - \Sigma_l^{-1}\boldsymbol{\mu}_l$$

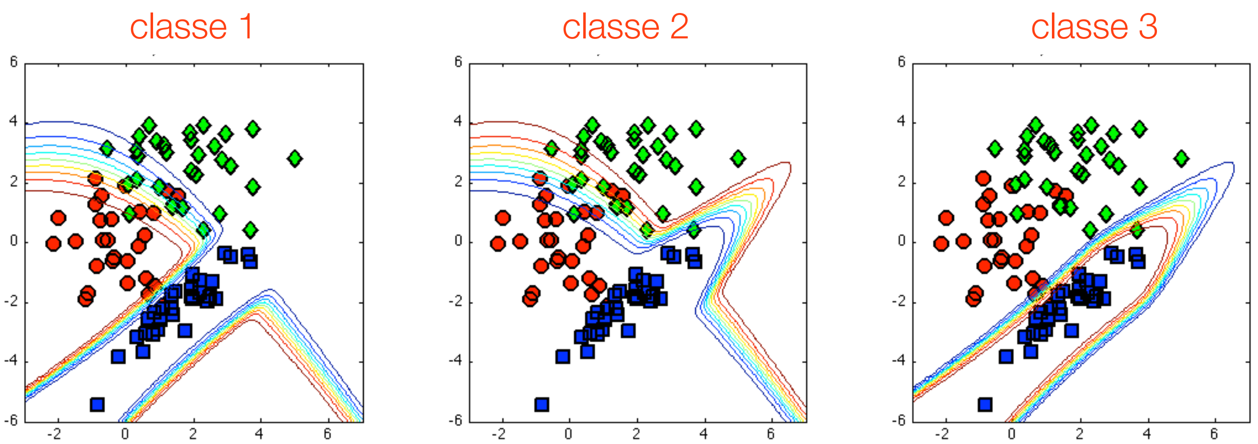
$$b_0 = \log \frac{P(C=k)}{P(C=l)} + \frac{1}{2}\log\frac{|\Sigma_l|}{|\Sigma_k|} + \frac{1}{2}\left(\boldsymbol{\mu}_l^\mathsf{T}\Sigma_l^{-1}\boldsymbol{\mu}_l - \boldsymbol{\mu}_k^\mathsf{T}\Sigma_k^{-1}\boldsymbol{\mu}_k\right)$$

Modelli generativi di classificazione
// Modello Gaussiano (GDA)

```
%% Load dei dati
load bc_data

% Plot dei dati

cl = unique(t);
col = {'ko','kd','ks'}
fcol = {[1 0 0],[0 1 0],[0 0 1]};
figure(1);
title('Dati originali')
hold off
for c = 1:length(cl)
    pos = find(t==cl(c));
    plot(X(pos,1),X(pos,2),col{c},...
        'markersize',10,'linewidth',2,...
        'markerfacecolor',fcol{c});
    hold on
end
xlim([-3 7])
ylim([-6 6])
```



Modelli generativi di classificazione
// Modello Gaussiano (GDA)



Modelli generativi di classificazione
// Naive Bayes (Bayes degli idioti)

- Adesso definiamo una funzione discriminante che tiene conto degli a priori sulle classi

$$\log \frac{P(C=1|\mathrm{x})}{P(C=0|\mathrm{x})} = \log \frac{P(\mathrm{x}|C=1)P(C=1)}{P(\mathrm{x}|C=0)P(C=0)}$$

modello della likelihood

- Funzione di verosimiglianza Gaussiana:

$$p(\mathrm{x}|C=k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left\{-\frac{1}{2}(\mathrm{x}-\boldsymbol{\mu}_k)^\mathsf{T}\Sigma_k^{-1}(\mathrm{x}-\boldsymbol{\mu}_k)\right\}$$

Matrice di covarianza diagonale

$$\Sigma_k = \begin{pmatrix} \sigma_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_3^2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & \sigma_{D-1}^2 & 0 \\ 0 & \cdots & \cdots & 0 & \sigma_D^2 \end{pmatrix}$$

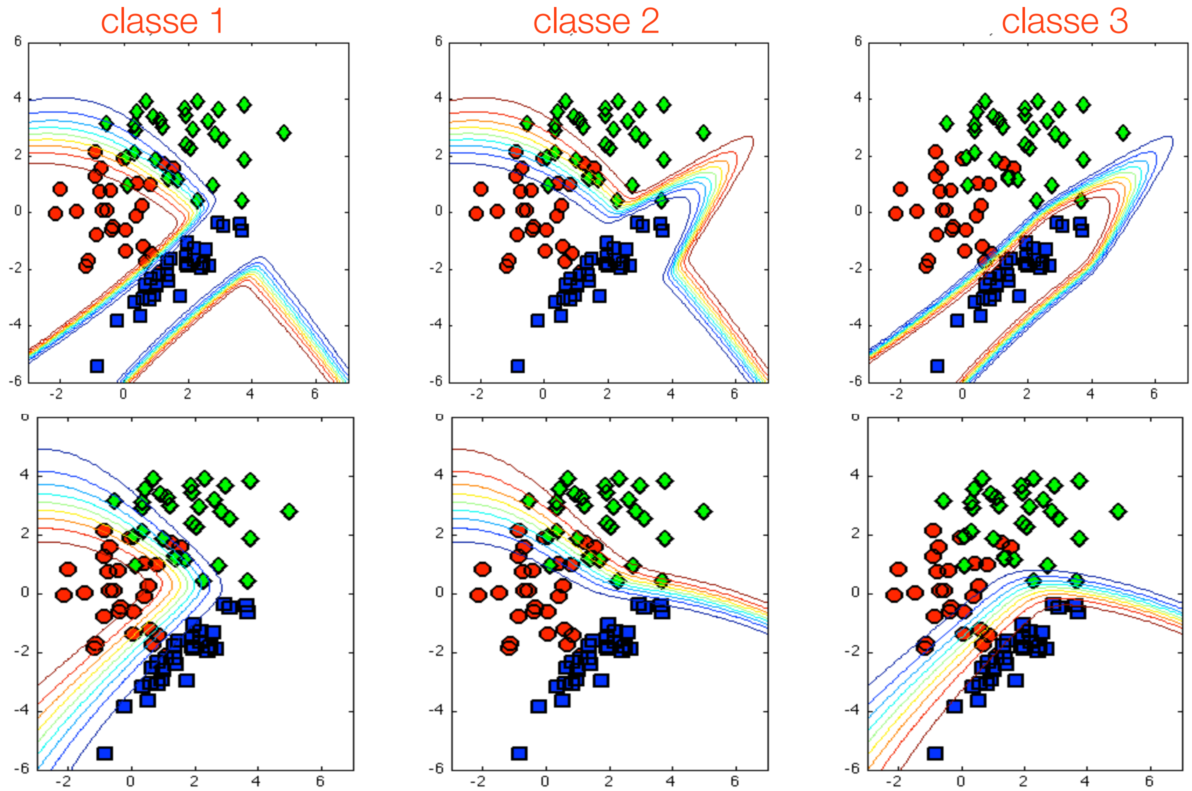
$$p(\mathrm{x}|C=k) = \prod_{d=1}^{D} p(x_d|C_k) = \prod_{d=1}^{D} \mathcal{N}_{x_d}(\mu_d, \sigma_d)$$

Modelli generativi di classificazione
// Naive Bayes (Bayes degli idioti)

$$\boxed{p(\mathrm{x}|C=k) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left\{-\frac{1}{2}(\mathrm{x}-\boldsymbol{\mu}_k)^\mathsf{T}\Sigma_k^{-1}(\mathrm{x}-\boldsymbol{\mu}_k)\right\}}$$

```
%  Stima con l'ipotesi Naive
for c = 1:length(cl)
    pos = find(t==cl(c));
    % Find the means
    class_mean(c,:) = mean(X(pos,:));
    class_var(c,:) = var(X(pos,:),1);
end


%% Probabilità predittive
[Xv,Yv] = meshgrid(-3:0.1:7,-6:0.1:6);
Probs = [];
for c = 1:length(cl)
    temp = [Xv(:)-class_mean(c,1) Yv(:)-class_mean(c,2)];
    sigma_k = diag(class_var(c,:));
    const = -log(2*pi) - log(det(sigma_k));
    Probs(:,:,c) = reshape(exp(const - 0.5*diag(temp*inv(sigma_k)*temp')),size(Xv));;
end

Probs = Probs./repmat(sum(Probs,3),[1,1,3]);
```

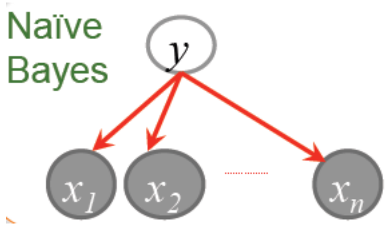
Modelli generativi di classificazione
// Modello Gaussiano vs Gaussiano Naive



Modelli generativi di classificazione
// Naive Bayes (Bayes degli idioti)

- Caso non Gaussiano: $x_i \in \{0,1\}$ Feature binarie
- Ipotesi: features indipendenti, data la classe

$$\boxed{p(\mathbf{x} \mid C_k) = \prod_i p(x_i \mid C_k)} = \prod_{i=1}^{D} \mu_{ki}^{x_i} (1-\mu_{ki})^{1-x_i}$$



- Regola di decisione: $C^* = \arg\max_k \, p(C_k) \prod_i p(x_i \mid C_k)$

- Per il fitting è ancora un GLM

$$a_k(\mathrm{x}) = \ln(p(\mathrm{x} \mid C_k) p(C_k)) = \sum_{i=1}^{D} \{x_i \ln \mu_{ki} + (1-x_i) \ln(1-\mu_{ki}\} + \ln p(C_k)$$

Lineare in x

- Esempio: classificazione di documenti (Information Retrieval)

Modelli generativi di classificazione
// Naive Bayes: Bag-of-Words

- Supponiamo di avere d documenti e un dizionario D di parole w.
- In ciascun documento la parola w puo' esserci o non esserci (bag of words)
  - Probabilita' che w sia presente o no in un documento di classe k (sport, Natale,...)

$$p_{kw} \qquad 1 - p_{kw}$$

parole

documento

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{dw} & & \\ & & \\ & & \end{pmatrix}$$

$$p(\mathbf{D}_d | C = k) = \prod_{w=1}^{|\mathcal{D}|} p(\mathbf{D}_{dw} | C_k) = \prod_{w=1}^{D} p_{kw}^{\mathbf{D}_{dw}} (1 - p_{kw})^{1-\mathbf{D}_{dw}}$$

Modelli generativi di classificazione
// Naive Bayes: Bag-of-Words

$$p(\mathbf{D}_d | C = k) = \prod_{w=1}^{|\mathcal{D}|} p(\mathbf{D}_{dw} | C_k) = \prod_{w=1}^{D} p_{kw}^{\mathbf{D}_{dw}} (1 - p_{kw})^{1-\mathbf{D}_{dw}}$$

- Stima dei parametri
  - ML

$$\widehat{p}_{kw} = \frac{1}{N_k} \sum_{d \in C_k} \mathbf{D}_{dw}$$

  - Bayesiana

$$\widehat{p}_{kw} = \frac{1 + \sum_{d \in C_k} \mathbf{D}_{dw}}{2 + N_k}$$

Modelli generativi di classificazione: GDA //input gaussiano vs LR

- GDA ha una forte assunzione di Gaussianità dei dati di input
  - se l'ipotesi è vera il classificatore è asintoticamente efficiente (il migliore)

- LR più robusta, meno dipendenza dalle ipotesi sui dati (gaussianità)
  - se input non gaussiano per grandi N, LR è migliore di GDA